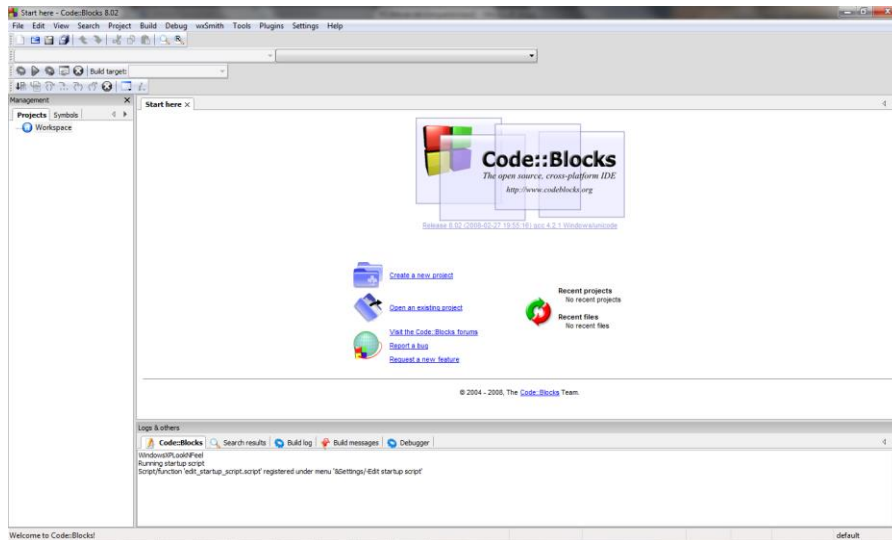


# PROGRAMACIÓN ALGORITMOS Y ESTRUCTURAS DE DATOS

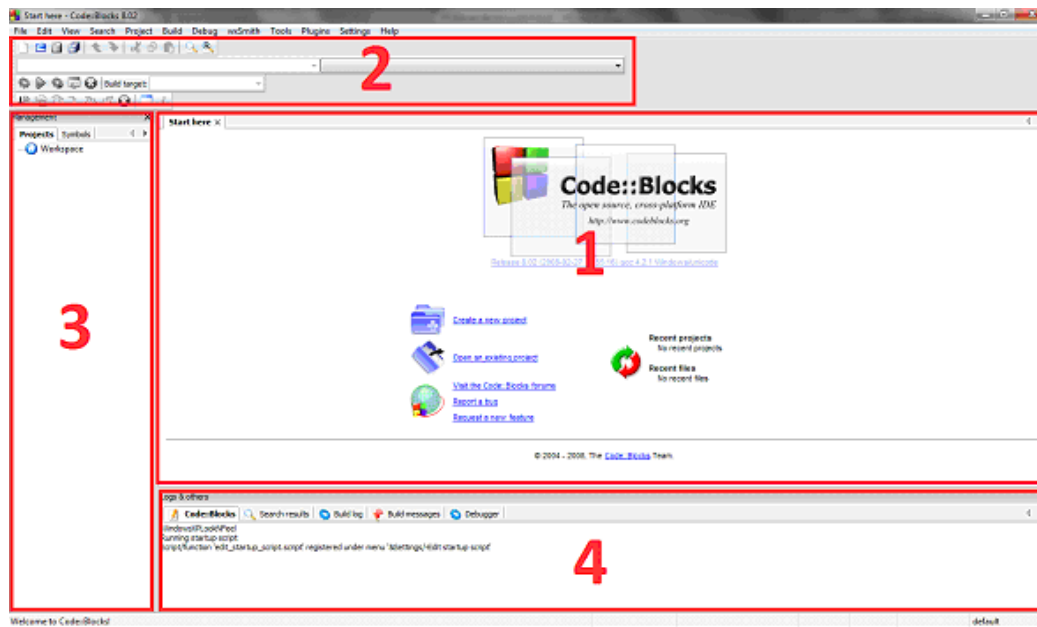
## TUTORIAL IDE CODEBLOCKS

### Entorno de Programación.

1. Encender la PC, la misma será gestionada por el Sistema Operativo Windows.  
Ingresar  
**Usuario :**  
**Password :**
2. Crear una carpeta de trabajo con el nombre Ej: **Grupo1**.  
Opciones:
  - a) Barra de Tareas->**Inicio->Programas->Explorador de Windows**  
Ubicar el lugar donde crear la carpeta . Ejemplo C:\  
**Archivo->Nuevo->Carpeta**
  - b) Tercer botón del mouse->**Nuevo->Carpeta**
3. Ejecutar la aplicación Codeblocks  
Barra de Tareas->**Inicio->Programas-> Codeblocks -> Codeblocks**
4. Aparece la ventana de presentación, y luego la pantalla principal del entorno.

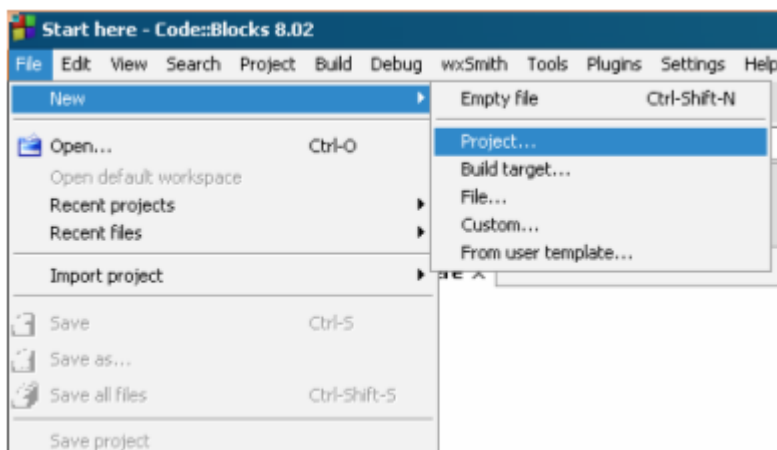


5. El ambiente de trabajo presenta 4 áreas principales:

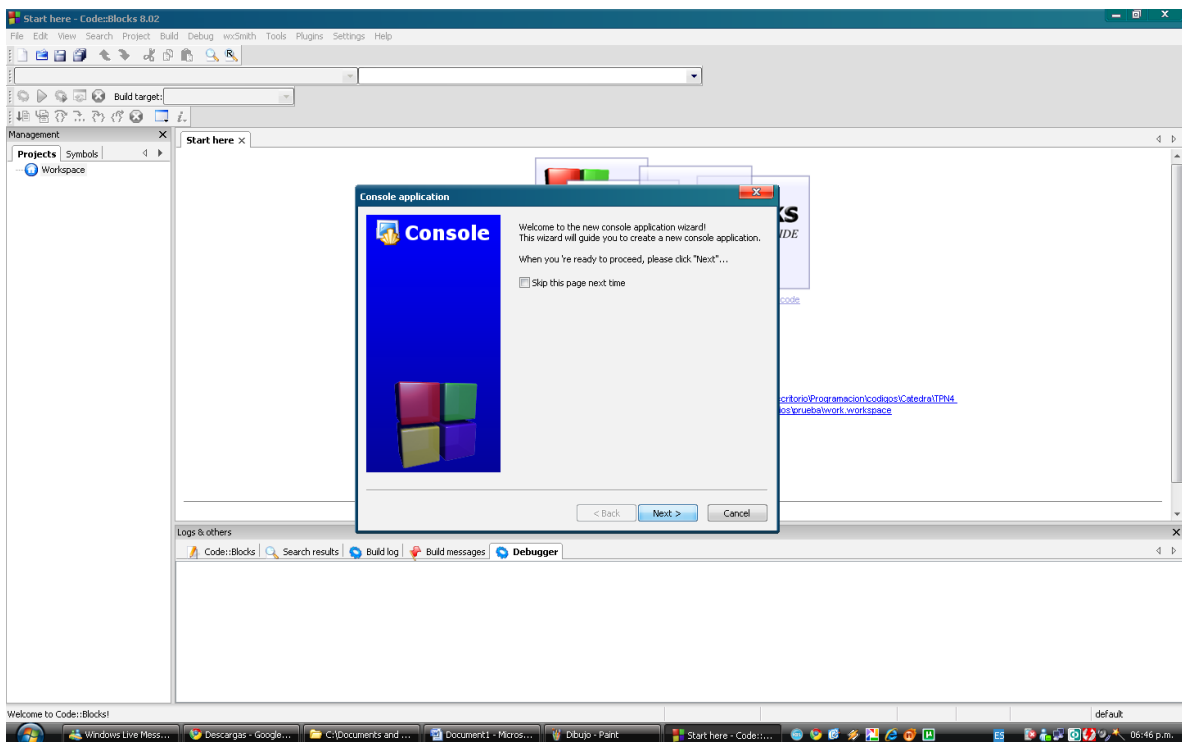
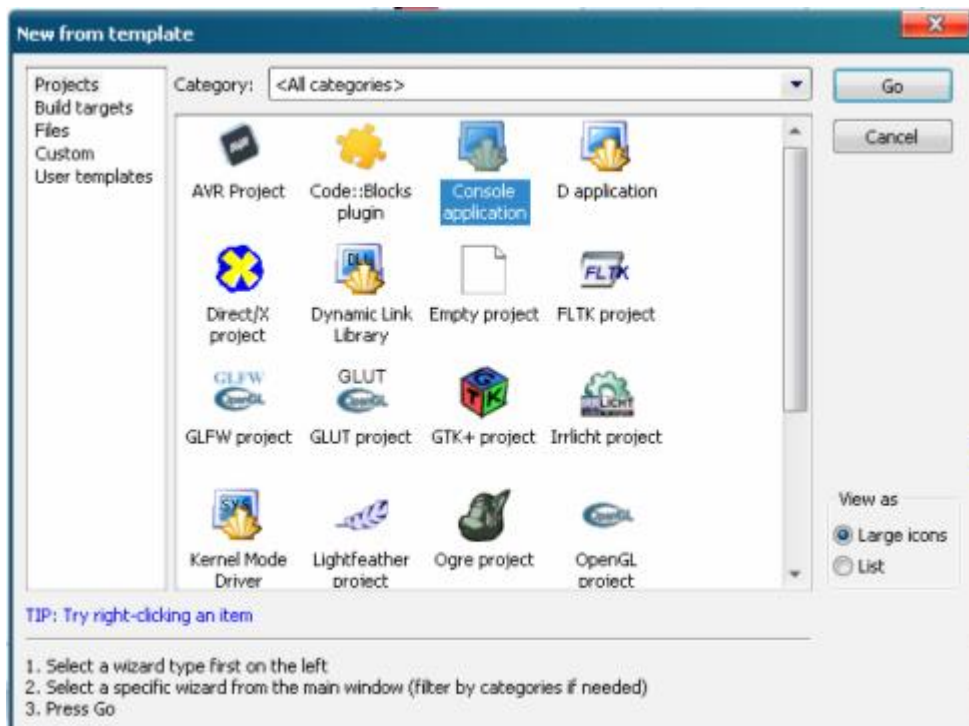


1. Área de trabajo y edición
2. Menú y barras de herramientas.
3. Explorador de proyectos.
4. Área de resultados.

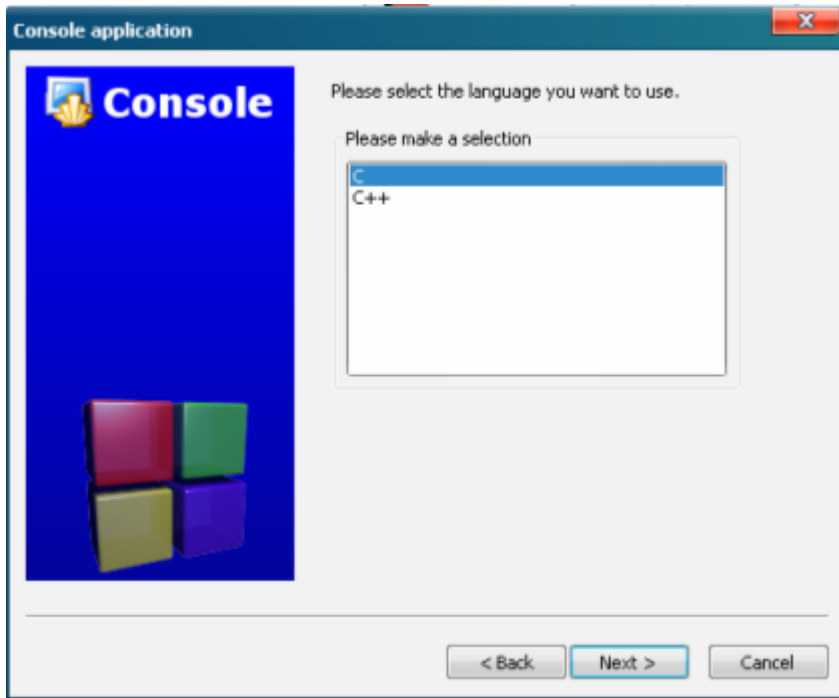
Luego se debe generar un nuevo proyecto como se muestra en la imagen de abajo:



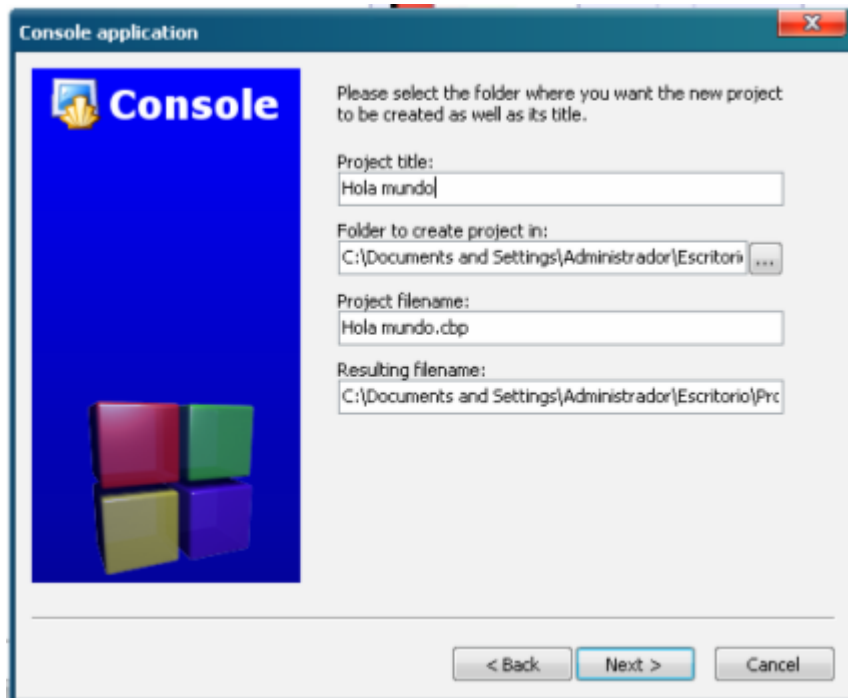
Elegir un template de aplicación de consola



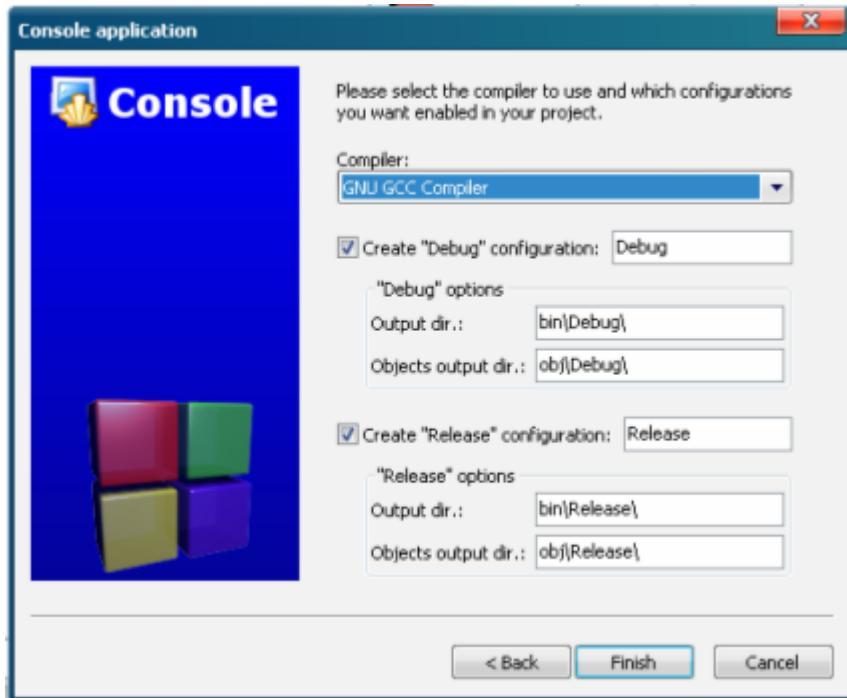
Elegir el lenguaje de programación “C”



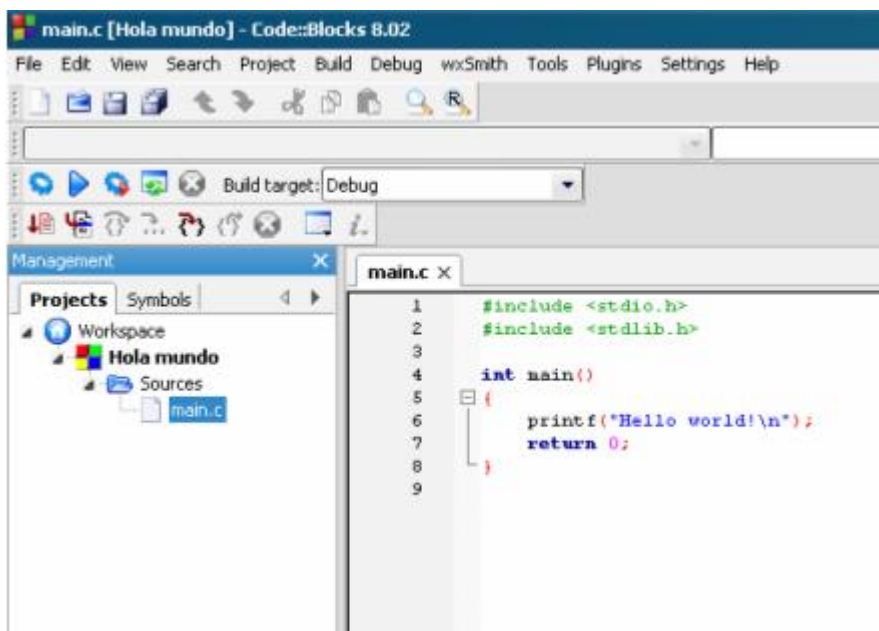
Definir un nombre del proyecto y una carpeta donde el mismo se alojará



Finalmente configurar el compilador como se muestra en la imagen de abajo (configuración por defecto).



Al presionar “Finalizar” aparecerá la siguiente ventana con el archivo **“main.c”** incluido en el proyecto y listo para compilar una aplicación de demostración. El texto de la ventana de la derecha puede ser modificado para generar código propio.



*Eliminar el código que se pone por defecto y reemplazarlo por el siguiente:*

```
#include <stdio.h>
int main(void)
{
    printf("Programación Algoritmos y Estructuras de Datos \n");
    getchar();
}
```

El editor usa diferentes colores en la sintaxis.

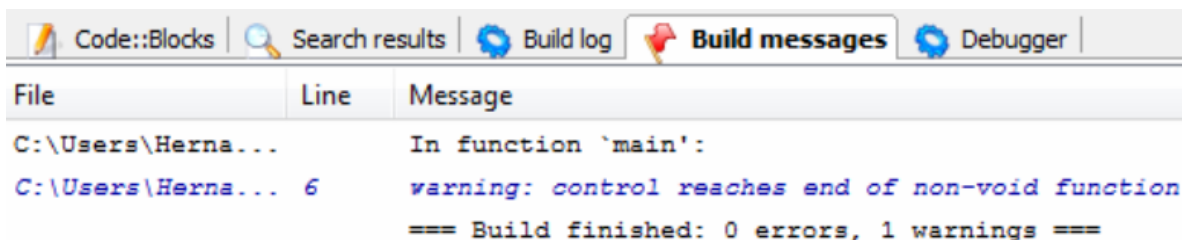
9. Compilar y ejecutar el programa

Opciones:

- a) **build->build**
- b) Icono build del menu

La línea verde al principio del archivo fuente es una directiva para el **preprocesador**. Este programa integrado al compilador, se ejecutará automáticamente previo proceso de compilación e incluirá en este caso el fichero *stdio.h*.

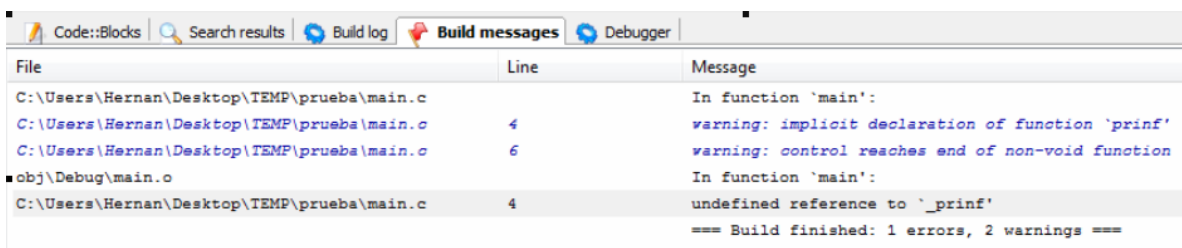
En la solapa de BUILD del “área de resultados” deberá aparecer una leyenda indicando que la compilación tuvo “0errores”



File	Line	Message
C:\Users\Herna...		In function 'main':
C:\Users\Herna... 6	6	warning: control reaches end of non-void function
		=== Build finished: 0 errors, 1 warnings ===

10. En caso de errores en el proceso de Compilación.

Aparecerá una lista de los mismos indicando la línea en la cual se encuentra el error y una breve descripción del mismo.



File	Line	Message
C:\Users\Hernan\Desktop\TEMP\prueba\main.c		In function 'main':
C:\Users\Hernan\Desktop\TEMP\prueba\main.c	4	warning: implicit declaration of function 'printf'
C:\Users\Hernan\Desktop\TEMP\prueba\main.c	6	warning: control reaches end of non-void function
obj\Debug\main.o		In function 'main':
C:\Users\Hernan\Desktop\TEMP\prueba\main.c	4	undefined reference to '_printf'
		=== Build finished: 1 errors, 2 warnings ===

11. Si no hay errores en el paso 10, verifique la existencia del programa ejecutable “*miprograma.exe*” en el directorio de trabajo\bin\debug.

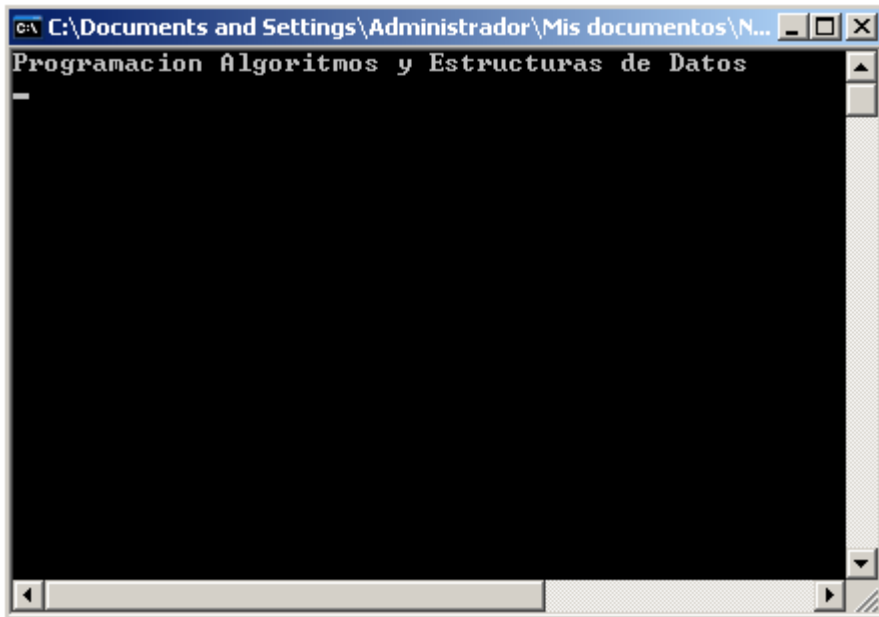
12. Ejecutar el programa “*miprograma.exe*”

Opciones:

a) **build->Run**

b) Pulsar sobre el Icono Run del menú.

Aparecerá entonces una ventana MS-DOS con el siguiente contenido:



Al pulsar una tecla la ventana se cerrará y volveremos al entorno de Codeblocks.

El programa queda a la espera que el usuario presione una tecla debido a la función de manejo de entrada de dato ***getchar()***. Si no existiera esta función la ventana mostrada arriba se cerraría al terminar la ejecución en forma instantánea.

### **Ayuda**

En el menú principal, se encuentra el ítem Ayuda, donde encontrará información sobre el entorno de programación y un tutorial básico de lenguaje C.